



Sistemi ICT per il Business Networking

Unified Modelling Language (UML) Class Diagram

Docente: **Massimo Cossentino**
Slide adattate dagli originali di: **V. Morreale**

Class Diagram/Object Diagram - 1

Argomenti da trattare

- Classi
- Oggetti
- Relazione tra classi ed oggetti: classificazione
- Identificazione degli oggetti
- Attributi (classi ed oggetti)
- Associazioni (classi)
 - Navigabilità delle associazioni
- Link (oggetti)
- Tecniche di individuazione delle classi

Classi in UML

- Una **classe** modella un insieme di entità (le istanze della classe) aventi tutti lo stesso tipo di caratteristiche (attributi, associazioni, operazioni...).
- Ogni classe è descritta da:
 - un **nome**
 - un insieme di **features**: attributi, operazioni, ...

Nome della classe e
"stereotipi"

"attributi"

"operazioni"

Estensioni UML



Tutti i comparti, a parte quello per il nome, possono essere non mostrati

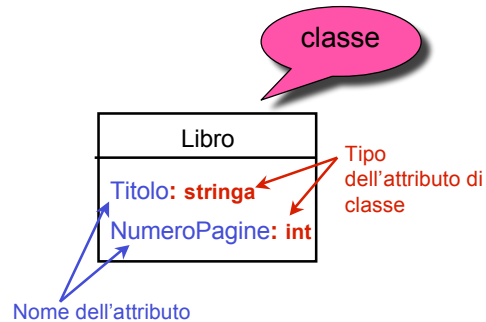
Attributi

- Attributo in una classe UML**
 - rappresenta un **attributo** posseduto da tutte le istanze della classe.
 - ha un **Nome** e un **Tipo**.
- La definizione di un attributo è **locale alla classe**, cioè un'altra classe può avere un attributo col medesimo nome, ma diversa definizione.

- Definizione:

**<visibilità><nome>: <tipo><molteplicità> = <default>
{<properties>}**

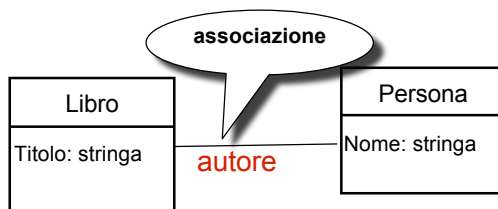
Attributi: esempio



Caratteristiche di classi: Associazioni

Una **associazione binaria** è rappresentata da una linea continua fra due classi. E' descritta da un **Nome**. Modella una relazione che può sussistere solo fra istanze della prima classe e istanze della seconda classe.

Significato analogo a quello dei diagrammi ER.

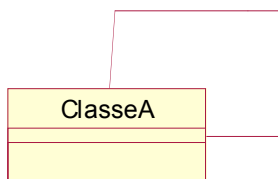


Significato:
autore è una relazione che può sussistere solo tra una persona ed un libro.

Nota: autore è una proprietà sia di libro sia di persona, cioè a differenza degli attributi le associazioni sono proprietà non locali.

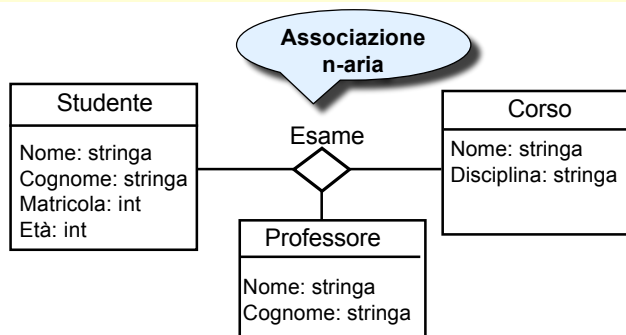
Associazioni (Cont.)

Un' **auto-associazione** è un'associazione tra una classe e se stessa



Associazioni (Cont.)

Una **associazione n-aria** è rappresentata da un rombo (**diamante**) connesso da **linea continue** a più classi. E' descritta da un **Nome**. Modella una relazione n-aria che **può sussistere solo** fra istanze delle classi connesse al diamante.

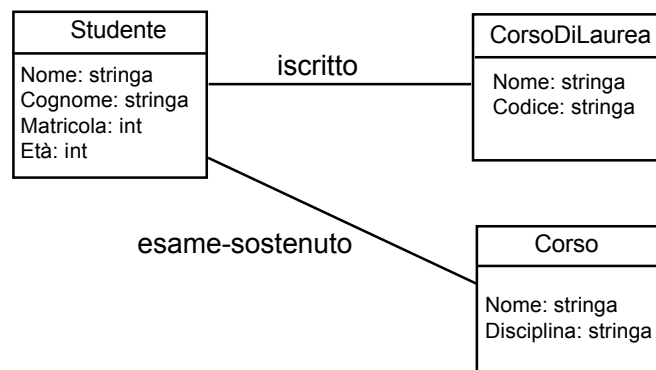


Esempio

- Tracciare il diagramma delle classi corrispondenti alle seguenti specifiche:

Si vogliono modellare gli studenti (con nome, cognome, numero di matricola, età), il corso di laurea in cui sono iscritti, ed i corsi di cui hanno sostenuto l'esame. Di ogni corso di laurea interessa il codice e il nome. Di ogni corso interessa il nome e la disciplina a cui appartiene (ad esempio: matematica, fisica, informatica, ecc.).

Esempio: Diagramma delle classi



Esercizio 2

- Tracciare il diagramma delle classi corrispondenti alle seguenti specifiche:

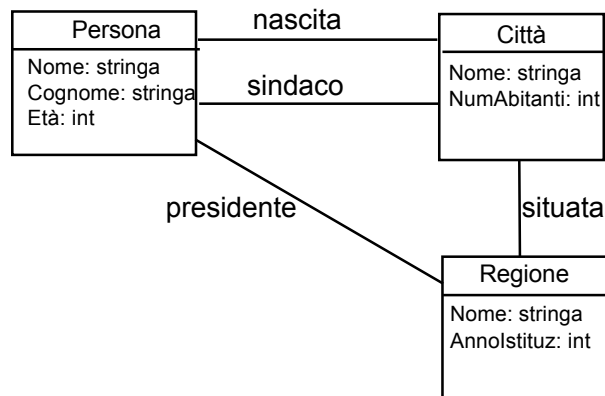
Si vogliono modellare le persone (con nome, cognome, età), le città di nascita (con nome, numero di abitanti, e sindaco), e le regioni in cui si trovano le città (con nome, anno di istituzione, e presidente).

Esercizio 2: soluzione

- Identificare i nomi:

Si vogliono modellare le persone (con nome, cognome, età), le città di nascita (con nome, numero di abitanti, e sindaco), e le regioni in cui si trovano le città (con nome, anno di istituzione, e presidente).

Esercizio 2: soluzione



22 ottobre 2007

ICT Systems for Business Networking

13

Class Diagram/Object Diagram - 1

- Classi
- Oggetti
- Relazione tra classi ed oggetti: classificazione
- Identificazione degli oggetti
- Attributi (classi ed oggetti)
- Associazioni (classi)
 - Navigabilit  delle associazioni
- Link (oggetti)
- Tecniche di individuazione delle classi

22 ottobre 2007

ICT Systems for Business Networking

14

Class Diagram/Object Diagram - 2

Argomenti da trattare

- Aggregazione e Composizione
- Dipendenza: alcune tipologie
- Verso dei nomi nelle associazioni
- Ruoli nelle associazioni
- Associazioni con attributi (association class)
 - Associazioni n-arie con attributi

Tipi particolari di associazione (Aggregazione e Composizione)

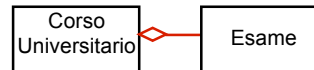
- **Aggregazione:**
 - è un associazione che corrisponde ad una relazione intuitiva Intero-Parte ("**part-of**")
 - è rappresentata da un **rombo vuoto** sull'associazione, posto vicino alla classe le cui istanze sono gli "interi"
 - si applica solo ad **associazioni binarie**
- **Composizione:**
 - È un aggregazione che rispetta due vincoli ulteriori:
 - una parte può essere inclusa in al massimo un intero in ogni istante.
 - solo l'oggetto intero può creare e distruggere le sue parti.
 - è rappresentata da un **rombo pieno** vicino alla classe che rappresenta gli "interi"



Tipi particolari di associazione (Aggregazione e Composizione)

■ **Aggregazione:**

- È una relazione **binaria**
- Può essere **ricorsiva**



■ **Composizione:**

- Se l'oggetto che compone viene distrutto, anche i figli vengono distrutti, ...
- ... anche se i figli possono essere creati/distrutti in momenti diversi dalla creazione/distruzione dell'oggetto che compone
- Può essere **ricorsiva**



Altre Relationship UML: **Dipendenza**

UML permette di rappresentare relazioni che, a differenza delle associazioni, **non sussistono fra istanze nel dominio rappresentato**, ma sussistono **fra gli elementi del modello UML stesso o fra le astrazioni che tali elementi rappresentano**.



■ **Dipendenza**

- è rappresentata da una linea tratteggiata orientata che va da un elemento *dipendente* ad uno *indipendente*
- Una dipendenza indica che **cambiamenti dell'elemento indipendente influenzano l'elemento dipendente** (cioè modificano il significato dell'elemento dipendente o causano la necessità di modificare anche l'elemento dipendente perché i significati sono dipendenti).

Alcune tipologie di dipendenza

- **call**: il Source chiama un'operazione sul Target
- **create**: il Source crea istanze del Target
- **derive**: il Source è derivato dal Target
- **instantiate**: il Source è un'istanza del Target (se il Source è una classe, il Target è una metaclassa)
- **permit**: il Target permette al Source di accedere alle sue proprietà private
- **realize**: il Source è un'implementazione di una specifica o interfaccia definita dal Target
- **refine**: rappresenta una relazione tra diversi livelli semantici (es. Source potrebbe indicare una classe al livello di design che corrisponde al Target al livello di analisi)
- **substitute**: il Source è sostituibile al Target
- **use**: il Source necessita dell'implementazione del Target

22 ottobre 2007

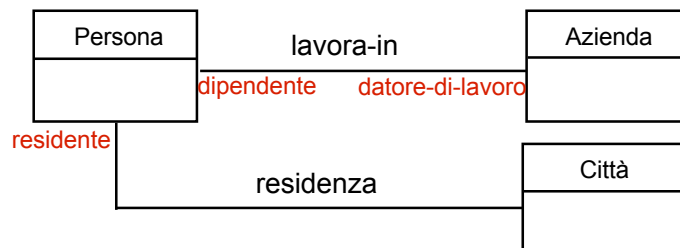
ICT Systems for Business Networking

19

Associazioni (ruoli)

Assegnare dei nomi ai "**ruoli**" svolti da ciascun elemento di un associazione.

Anche nei casi in cui non è strettamente necessario, il ruolo può essere utile per aumentare la leggibilità del diagramma



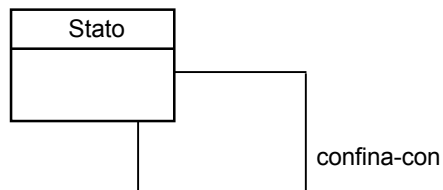
22 ottobre 2007

ICT Systems for Business Networking

20

Associazioni (ruoli)

- Ci sono casi in cui l'associazione **insiste** più volte sulla stessa classe, ma il ruolo non è significativo, in particolare quando l'associazione rappresenta una **relazione simmetrica**



22 ottobre 2007

ICT Systems for Business Networking

21

Class Diagram/Object Diagram - 2

- Aggregazione e Composizione
- Dipendenza: alcune tipologie
- Verso dei nomi nelle associazioni
- Ruoli nelle associazioni
- Associazioni con attributi (association class)
 - Associazioni n-arie con attributi

22 ottobre 2007

ICT Systems for Business Networking

22

Esercitazione

Modellare con un diagramma delle classi i vari tipi di cellule che esistono nel corpo umano:

- Cellule sanguigne, cellule nervose e cellule muscolari
- Le cellule sanguigne possono essere globuli rossi o bianchi
- Le cellule muscolari possono essere lisce o striate
- Le cellule nervose possono essere piramidali o corticali

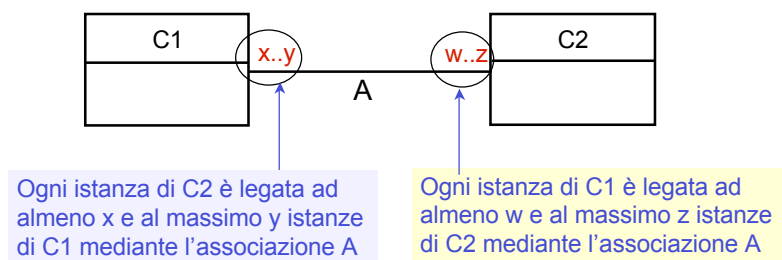
Class Diagram/Object Diagram - 3

Argomenti da trattare

- Molteplicità nelle associazioni
 - Associazioni n-arie con molteplicità
- Molteplicità degli attributi
- Attributi vs. associazioni
- La relazione "IS A"
 - Classificazione
 - Generalizzazione: Partizioni, Ereditarietà multipla, Disjoint/Overlapping, Generalizzazione completa
- Classi astratte
- Generalizzazione tra associazioni
- Overriding tra attributi

Associazioni (molteplicità)

- Per specificare con maggiore precisione il significato delle **associazioni** si possono definire i vincoli di molteplicità (o semplicemente **molteplicità**).

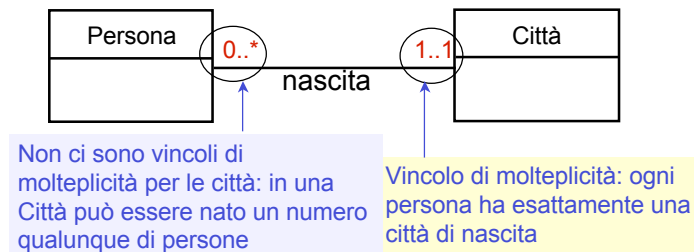


Associazioni (molteplicità)

- Data l'espressione di molteplicità $[n..m]$, si evince che i seguenti modi alternativi di esprimerla corrispondono a:
 - Opzionale** → $n = 0$
 - Obbligatorio** → $n = 1$
 - Valore singolo** → $m = 1$
 - Multivalore** → $m = *$
- Il **valore di default** della molteplicità è **[1]**.
- È **preferibile** comunque esprimere sempre la molteplicità, perché spesso l'omissione deriva dalla volontà di non evidenziare tale dettaglio, piuttosto che dall'assunzione del valore di default

Esempio (associazioni - molteplicità)

- Ogni istanza di Persona deve essere legata ad esattamente una istanza (cioè ad almeno una e al massimo una) di Città da link dell'associazione "nascita".



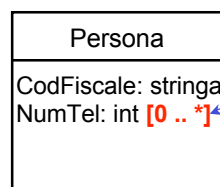
22 ottobre 2007

ICT Systems for Business Networking

27

Attributi (molteplicità)

- Si può specificare anche la **molteplicità degli attributi**, fra parentesi quadre dopo il tipo. Il vincolo di default è **[1..1]**.



Molteplicità di attributo:
una persona può avere un numero qualunque di numeri di telefono

*Un attributo di tipo T della classe C con molteplicità diversa da $[1..1]$ si dice **multivalore** e formalmente è una relazione tra la classe C ed il tipo T*

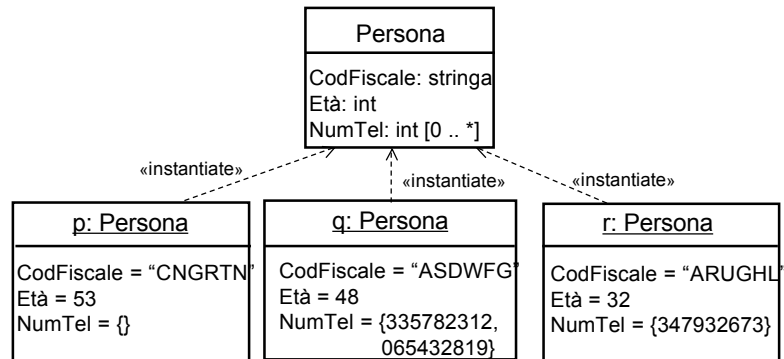
22 ottobre 2007

ICT Systems for Business Networking

28

Attributi (molteplicità)

- Nelle istanze, il **valore di un attributo multivalore** si indica mediante un **insieme**



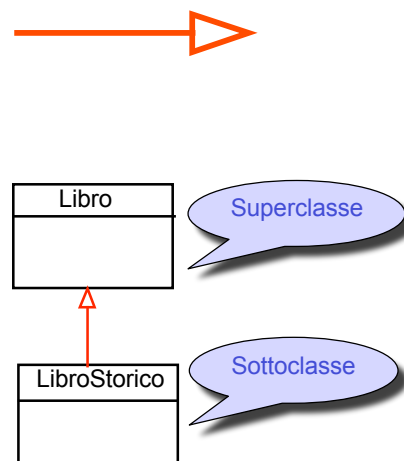
22 ottobre 2007

ICT Systems for Business Networking

29

Altre Relationship UML: Generalizzazione

- La generalizzazione è indicata con una freccia vuota fra due classi dette **sottoclasse** e **superclasse**.
- Il significato della generalizzazione è il seguente: **ogni istanza della sottoclasse è anche istanza della superclasse**.



22 ottobre 2007

ICT Systems for Business Networking

30

Altre Relationship UML: Generalizzazione

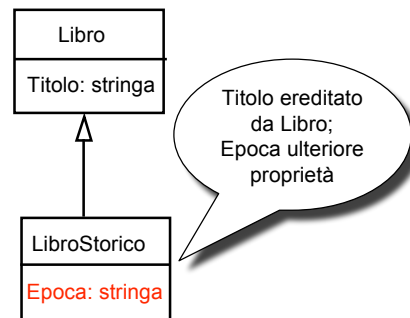
Principio di ereditarietà: ogni proprietà (attributi, associazioni, operazioni) della superclasse è anche una proprietà della sottoclasse e non si riporta esplicitamente nel diagramma (a meno di overriding)

Dal fatto che

1. ogni istanza di Libro ha un Titolo
2. ogni istanza di LibroStorico è una istanza di Libro

segue logicamente che

3. ogni istanza di LibroStorico ha un Titolo



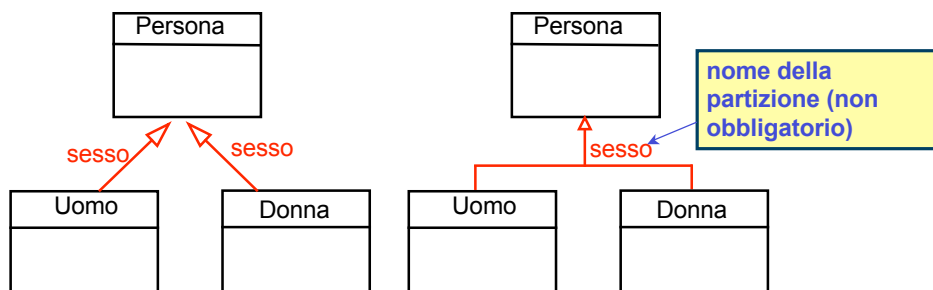
22 ottobre 2007

ICT Systems for Business Networking

31

Altre Relationship UML: Generalizzazione (partizioni)

- Le generalizzazioni possono essere raggruppate in **partizioni**. Di default le generalizzazioni appartengono alla partizione detta "default".
- Il nome della partizione (o del criterio di partizione) si scrive accanto alla generalizzazione. La label del nome si chiama anche "**discriminator**".
- Generalizzazioni appartenenti alla stessa partizione possono essere fuse in **un'unica freccia ramificata**.

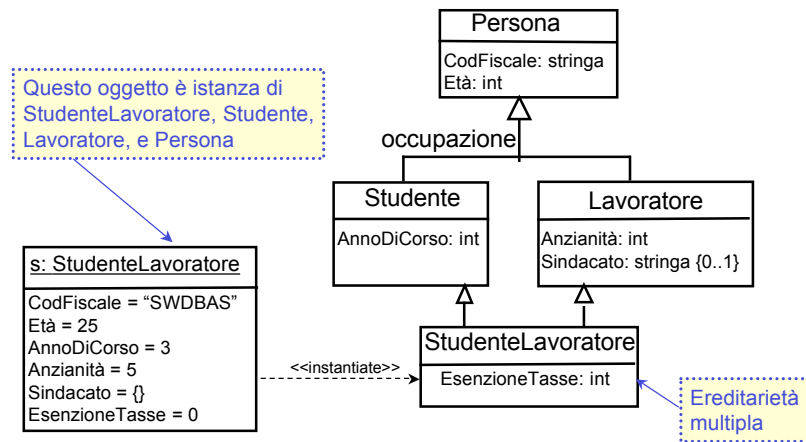


22 ottobre 2007

ICT Systems for Business Networking

32

Esempio (Generalizzazione-ereditarietà multipla)



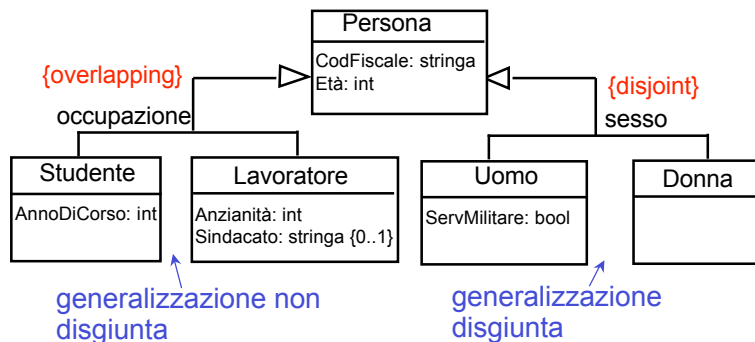
22 ottobre 2007

ICT Systems for Business Networking

33

Altre Relationship UML: Generalizzazione (disjoint / overlapping)

- La stessa superclasse pu  partecipare a diverse generalizzazioni
- Una generalizzazione pu  essere **disgiunta**, cio  le sottoclassi sono disgiunte (non hanno istanze in comune), o no.

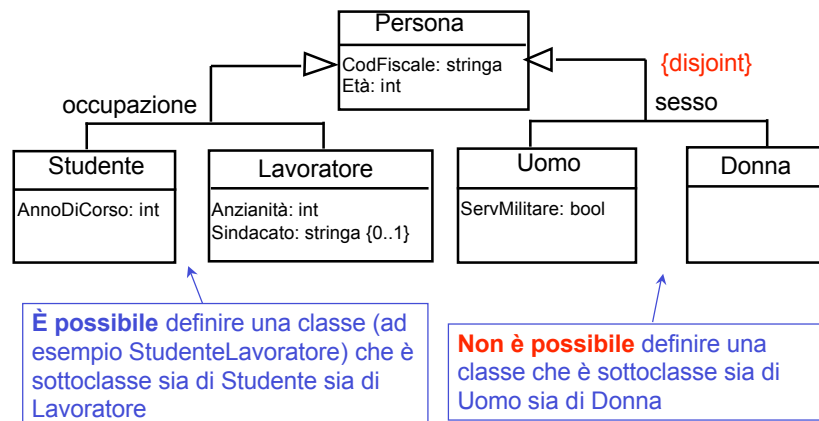


22 ottobre 2007

ICT Systems for Business Networking

34

Esempio (Generalizzazione-disjoint)



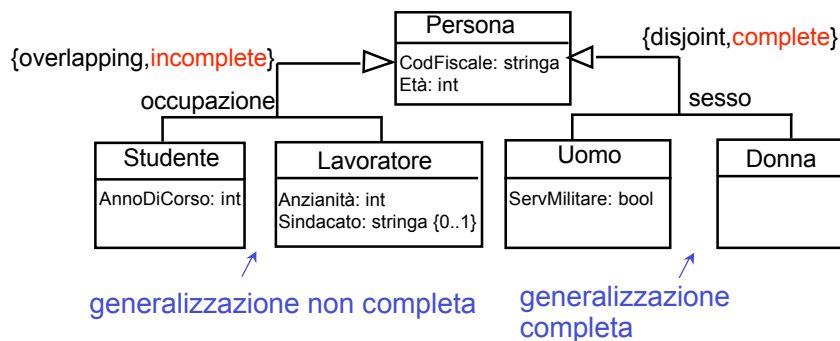
22 ottobre 2007

ICT Systems for Business Networking

35

Altre Relationship UML: Generalizzazione (complete / incomplete)

- Una generalizzazione può essere **completa** (l'unione delle istanze delle sottoclassi è uguale all'insieme delle istanze della superclasse) o no.
- Attenzione:** I **valori di default** sono **{incomplete, disjoint}**



22 ottobre 2007

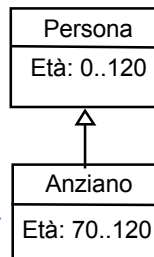
ICT Systems for Business Networking

36

Altre Relationship UML: Generalizzazione (overriding)

- In una generalizzazione la sottoclasse non solo può avere caratteristiche aggiuntive rispetto alla superclasse, ma può anche **sovrascrivere (overriding)** le proprietà ereditate dalla superclasse.

Overriding
dell'attributo Età



In UML 2.0 il termine "overriding" è sostituito da "ridefinizione" (redefining).

Eventuali vincoli fra proprietà della superclasse e proprietà corrispondente nella sottoclasse dipendono dal significato che le proprietà assumono nel particolare contesto di applicazione.

Ad es. spesso si assume questo vincolo: "un attributo può essere ridefinito solo sostituendo il tipo con un sottotipo".

Introduzione

- Cos'è una metodologia di sviluppo software
- Modelli e linguaggi di modellazione
- Alcune note sull'Object-Orientation (OO)
- UML: cos'è (e cosa non è)
- Meccanismi di estensione di UML: introduzione

Class Diagram/Object Diagram - 1

- Classi
- Oggetti
- Relazione tra classi ed oggetti: classificazione
- Identificazione degli oggetti
- Attributi (classi ed oggetti)
- Associazioni (classi)
 - Navigabilità delle associazioni
- Link (oggetti)
- Tecniche di individuazione delle classi

Class Diagram/Object Diagram - 2

- Aggregazione e Composizione
- Dipendenza: alcune tipologie
- Verso dei nomi nelle associazioni
- Ruoli nelle associazioni
- Associazioni con attributi (association class)
 - Associazioni n-arie con attributi

Class Diagram/Object Diagram - 3

- Molteplicità nelle associazioni
 - Associazioni n-arie con molteplicità
- Molteplicità degli attributi
- Attributi vs. associazioni
- La relazione "IS A"
 - Classificazione
 - Generalizzazione
 - Partizioni
 - Ereditarietà multipla
 - Disjoint/Overlapping
 - Generalizzazione completa
- Classi astratte
- Generalizzazione tra associazioni
- Overriding tra attributi

22 ottobre 2007

ICT Systems for Business Networking

41

Esercitazione

- Modellare con un diagramma delle classi il seguente sistema:
 - Un libro è composto di capitoli, un indice e una lista di parole chiave.
 - Un capitolo comprende diversi paragrafi, ognuno dei quali comprende diversi capoversi e figure.
 - Un capoverso comprende diverse frasi, ognuna delle quali comprende diverse parole.

22 ottobre 2007

ICT Systems for Business Networking

42

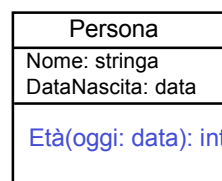
Class Diagram/Object Diagram - 4

Argomenti da trattare

- Operazioni
- Overriding tra operazioni
- Visibilità
- Commenti in UML
- Consigli sull'utilizzo dei Class Diagram

Operazioni

- Finora abbiamo visto proprietà "**strutturali**" (attributi e associazioni) che definiscono cosa gli oggetti sono. Le classi (e quindi le loro istanze) sono caratterizzate anche da proprietà "**comportamentali**" che definiscono cosa gli oggetti fanno: le **operazioni**.
- Una operazione associata ad una classe C rappresenta un **servizio** che può essere chiesto a tutte le istanze della classe.
- Un operazione ha
 - Un **nome**
 - Una lista di **argomenti**
 - Un eventuale **risultato**

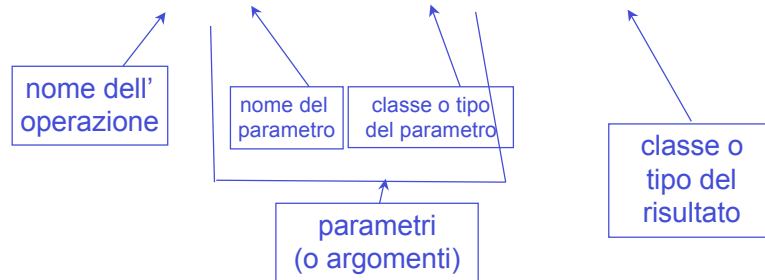


Operazione per chiedere l'età della persona

Operazioni (definizione)

■ Sintassi di un operazione

<visibilità> <nome> (X: T1, ... , Xn: Tn): <tipo-output> {<properties>}



22 ottobre 2007

ICT Systems for Business Networking

45

Operazioni: commenti

Attenzione: le operazioni che si definiscono sul modello di **analisi** sono le operazioni che caratterizzano concettualmente la classe. Altre operazioni, più orientate alla realizzazione del software (come ad esempio le operazioni che consentono di gestire gli attributi, ossia conoscerne o cambiarne il valore), non devono essere definite in questa fase

Operazioni vs. Metodi: per **operazione** si intende la dichiarazione della procedura; il **metodo** è il corpo della procedura.

Attenzione. Spesso vengono utilizzati come sinonimi.

22 ottobre 2007

ICT Systems for Business Networking

46

Consigli sull'utilizzo dei Class Diagram

- **Non usare sempre tutti i costrutti disponibili:** quando non necessario, appesantisce soltanto la notazione
- Utilizzare i Class Diagram per modellare **oggetti del dominio** e come mezzo di comunicazione → Notazione semplice
- Non modellare ogni cosa: concentrati sulle **cose importanti**
- Non ignorare gli aspetti dinamici: utilizza i Class Diagram insieme a qualche diagramma che fornisce una **vista dinamica** del sistema (es. Sequence Diagram, ...)
- Utilizzare **nomi al plurale** per gli attributi e le associazioni con molteplicità > 1

22 ottobre 2007

ICT Systems for Business Networking

47

Class Diagram/Object Diagram - 4

- Operazioni
- Overriding tra operazioni
- Visibilità
- Commenti in UML
- Consigli sull'utilizzo dei Class Diagram

22 ottobre 2007

ICT Systems for Business Networking

48

Esercitazione

- **Modellare il seguente sistema:**
 - Una automobile è identificata da un numero di telaio, anno di fabbricazione, produttore, modello, colore e peso. L'automobile può essere acquistata, venduta, guidata.
 - L'automobile ha un motore che eroga una certa potenza ed ha una determinata cilindrata. Il motore può essere spento e acceso.
 - Il motore è collegato ad una batteria caratterizzata da una data tensione ed energia. La batteria può essere caricata o scaricata (durante l'uso ad esempio)
 - L'automobile ha da 3 a 5 ruote di un certo diametro.
 - L'automobile ha un numero di porte variabile tra 2 e 5. Le porte possono essere aperte o chiuse

Esercitazione

- Esercizio 3
- **Esercizio 4**
- **Esercizio 5**

Esercitazione

- Esercizio 6
- **Esercizio 10**

22 ottobre 2007

ICT Systems for Business Networking

51

Materiale didattico

- **UML Distilled** (Capitoli relativi a Use Cases, Use Case Diagram, Class Diagram, Activity Diagram)

22 ottobre 2007

ICT Systems for Business Networking

52

Altri riferimenti

22 ottobre 2007

ICT Systems for Business Networking

53